

# AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method for visualizing any architecture ~~during conceptual, development and deployment phases~~ of a system, said method comprising the steps of:  
 receiving information regarding an event;  
 determining a look of an application;  
 determining a feel of the application;  
 receiving events at an event service delivery agent;  
 forwarding the events to a model component of the architecture in the form of callbacks;  
 reading configuration information during initialization at application startup, wherein the configuration information is adaptable to a plurality of different middlewares; ~~and~~  
~~presenting the associated behavior of the system.~~  
visualizing the architecture of the system during conceptual, development and deployment phases of the system;  
wherein the step of visualizing during the conceptual phase of the system is performed in a simulation mode to communicate architectural components and collaborations of the system before the architecture has been implemented in the development and deployment phases;  
wherein the step of visualizing during the development phase of the system is performed in either the simulation mode for sections of the system that have not yet been implemented or a mode for monitoring a deployed system for portions of the system that have already been implemented; and  
wherein the step of visualizing during the deployment phase of the system is performed in the mode for monitoring the deployed system.

2-3 Canceled

4. (Currently Amended) The method of claim 1, wherein said step of visualizing ~~presenting~~ comprises the step of visualizing ~~presenting~~ a deployed implementation of the architecture.

5. (Currently Amended) The method of claim 1, wherein said configuration information further comprises:  
 abstract information in the form of tiers, components, communication paths and events[[]];  
~~presentation~~ visualizing information in the form of how many display views are required to ~~present~~ visualize the architecture, and how to respond visually when events are received;  
 controller information that may specify details that determine how the particular controller implementation behaves; and  
 integration information that may be used by the particular implementation of an event service delivery agent.

6. (Currently Amended) An architecture visualization system for visualizing any architecture during conceptual, development or deployment phases of a system, said architecture visualization system comprising:

- a model component for receiving information regarding events;
- a view component for determining a look of an application;
- a controller for effectively determining a feel of the application;
- an event service delivery agent for receiving and delegating requests; and
- a configuration specified in XML, wherein the configuration is adaptable to a plurality of different middlewares;-

wherein the visualization is performed during conceptual, development and deployment phases of the architecture visualization system;

wherein the step of visualizing during the conceptual phase of the architecture is performed in a simulation mode to communicate architectural components and collaborations before the architecture has been implemented in the development and deployment phases;

wherein the step of visualizing during the development phase of the architecture is performed in either the simulation mode for sections of the system that have not yet been implemented or a mode for monitoring a deployed system for portions of the system that have already been implemented; and

wherein the step of visualizing during the deployment phase of the system is performed in the mode for monitoring the deployed system.

7. (Currently Amended) A system for visualizing an application architecture, the system comprising:

- an event service interface for receiving middleware or protocol dependent events, the event service interface for translating the middleware or protocol dependent events into middleware or protocol independent events;

- an event service delivery agent for receiving the middleware or protocol independent events and managing delivery of events to an application;

- a model component for receiving middleware or protocol independent events from the event service delivery agent;

- a view component receiving middleware or protocol independent events from the model component, the view component for visualizing behavior of an application during at least one of conception, development, and deployment of the application; and

- a controller for managing the behavior of the application, an implementation of the controller being adaptable based on the middleware communicating with the application;-

wherein the step of visualizing the behavior of the application during the conception of the application is performed in a simulation mode to communicate architectural components and collaborations before the development and deployment of the application architecture;

wherein the step of visualizing the behavior of the application during the development of the application is performed in either the simulation mode for sections of the application architecture that have not yet been implemented or a mode for monitoring a deployed system for portions of the application architecture that have already been implemented; and

wherein the step of visualizing the behavior of the application during the deployment of the application is performed in the mode for monitoring the deployed system.

8. (Previously Presented) The system of claim 7, further comprising an XML configuration for creating implementations of the event service delivery agent and the controller, wherein the implementations are based on the middleware communicating with the application.

9. (Previously Presented) The system of claim 8, wherein the controller manages behavior of the application based on at least one of information received from the view component, the model component, and XML configuration information related to the controller.

10. (Previously Presented) The system of claim 8, wherein the XML configuration comprises:

- abstract information in the form of tiers, components, communication paths and events;
- presentation information in the form of how many display views are required to present the architecture, and how to respond visually when events are received;
- controller information that may specify details that determine how a particular controller implementation behaves; and
- integration information that may be used by a particular implementation of an event service delivery agent.

11. (Previously Presented) The system of claim 7, wherein the view component creates different views of the application dependent on whether the system is in a conception mode, a development mode, or a deployment mode.

12. (Previously Presented) The system of claim 7, wherein the view component deploys a slowed down event visualization in a demonstration mode.

13. (Previously Presented) The system of claim 7, wherein the view component creates a visualization based on configuration information.

14. (Previously Presented) The system of claim 7, wherein the event service delivery agent forwards events to the model component as a callback instruction.